Nolan Stevenson
109841202
ASEN 6055
HW1

Question 1

1. Show that $E[\mathbf{w}] = \boldsymbol{\mu}$ and $E[(\mathbf{w} - \boldsymbol{\mu})(\mathbf{w} - \boldsymbol{\mu})^T] = \boldsymbol{\Sigma}$.

Given $w = \mu + Su$ Take expectations

$$E[w] = E[\mu + Su] = E[\mu] + S\,E[u]$$

μ is vector so $E[\mu] = \mu$ S is matrix so take out

and we know $u \sim N(0, I)$

$\therefore E[u] = 0$

Giving $\boxed{E[w] = \mu}$

For cov, $E[(w-\mu)(w-\mu)^T] = \Sigma$

$$w = \mu + Su, \quad w - \mu = Su$$

Form outer product

$$(w-\mu)(w-\mu)^T = (Su)(Su)^T = S(uu^T)S^T$$

Take expectation

$$E[(w-\mu)(w-\mu)^T] = E[S(uu^T)S^T] = S\,E[uu^T]S^T$$

w/ $u \sim N(0, I)$, $E[uu^T] = I$

Thus,

$$= S\,E[uu^T]S^T = S\,I\,S^T = SS^T$$

$$\therefore \boxed{E[(w-\mu)(w-\mu)^T] = SS^T = \Sigma}$$

2. Generate *iid* multivariate normal random fields on one-dimensional discrete locations, and estimate the sample mean and covariance from simulated random fields with sample size of 100, 1000, and 10000. Use the covariance matrix $\mathbf{\Sigma} \in \mathbb{R}^{100\times100}$ whose elements are filled with a parametric function $C_d = \sigma^2 exp(-d/\rho)$ that depends on the pair-wise distance between locations d and parameters σ and ρ. You may choose the values of σ and ρ as well as $\boldsymbol{\mu}$ for your exercise. Repeat the same exercise with a different covariance kernel function, for example, $C_{d^2} = \sigma^2 exp(-(d/\rho)^2)$.

We can simulate both kernals in Python with:

```python
# Locations - r_i = 0, 0.01, ..., 0.99
n_locations = 100
locations = np.linspace(0.0, 0.99, n_locations)

# Parameters
sigma2 = 1.0   # variance
rho = 0.1      # length-scale parameter
mu = np.zeros(n_locations)  # Zero means

# Pairwise distance matrix
D = np.abs(locations[:, None] - locations[None, :])

# Covariance kernels
def cov_exp(D, sigma2=1.0, rho=0.1):
    return sigma2 * np.exp(-D / rho)

def cov_exp_squared(D, sigma2=1.0, rho=0.1):
    return sigma2 * np.exp(- (D / rho)**2)
```

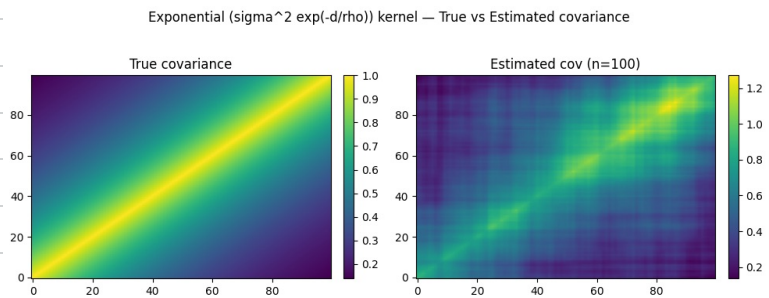
for 100 true samples and $\sigma = 1$, $\rho = 0.5$

3. Compare the sample covariances estimated from simulated random fields to its theoretical (true) covariance by visualizing the covariance surfaces at a given location. Discuss the sample mean and covariance uncertainty in terms of the multivariate central limit theorem.

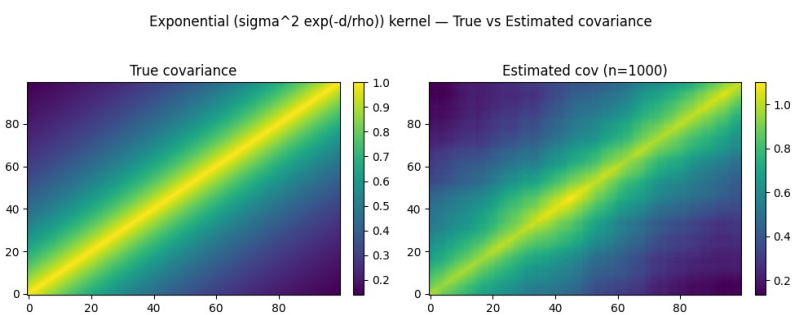
By simulating samples randomly from true distro we can compare accuracy w/ diff sample sizes

$$C_d = \sigma^2 \exp(-d/\rho)$$

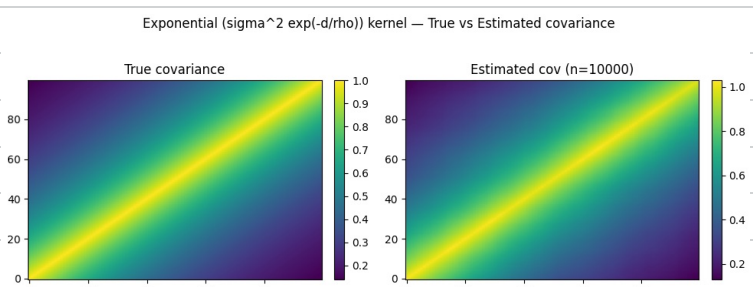
n=100:



n=1000:

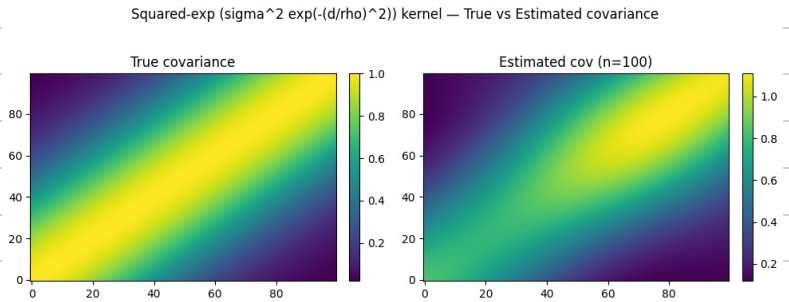


n=10000:

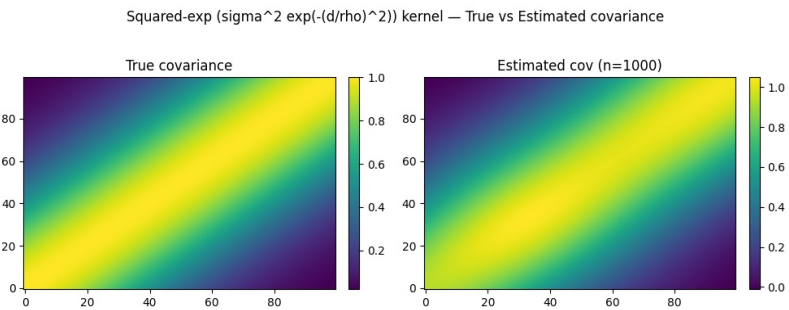


$C_d = \sigma^2 \exp(-(d/\rho)^2)$

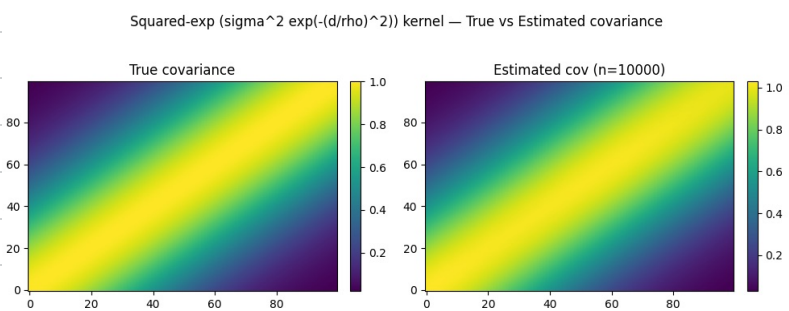
n=100:



n=1000:



n=10000:



We converge to the kernal as $n \uparrow$

Significant improvement from $n = 100$

Considering the mean and covariance from the samples, we have:

```
=== Kernel: Exponential (sigma^2 exp(-d/rho)) ===
n = 100: mean (first 5) = [-0.03551742 -0.04195332 -0.03052949 -0.0283869  -0.01226099]
n = 100: top-left 3x3 of sample covariance:
[[0.87748114 0.85420497 0.82913464]
 [0.85420497 0.86530333 0.84670443]
 [0.82913464 0.84670443 0.86873751]]
n = 1000: mean (first 5) = [-0.02083845 -0.00943948 -0.00419456 -0.01528724 -0.02400125]
n = 1000: top-left 3x3 of sample covariance:
[[0.98250314 0.95413993 0.93950684]
 [0.95413993 0.96700045 0.95277681]
 [0.93950684 0.95277681 0.97545487]]
n = 10000: mean (first 5) = [-0.00602848 -0.00662519 -0.0085514  -0.00979402 -0.01206537]
n = 10000: top-left 3x3 of sample covariance:
[[1.01868161 0.99962649 0.98103756]
 [0.99962649 1.0206163  1.00100678]
 [0.98103756 1.00100678 1.02037907]]
```

Where mean was taken along the row axis and printed for first 5 columns.

As you can see, following central limit theorm, as n↑, from 100 to 10,000 our mean per col converges to ≈0, from 0.0X to 0.00X. Factors smaller. This aligns w/ central limit thm.

Problem 2

1. Estimate the posterior distribution of $\mathbf{x}$. Suppose $\mathbf{x}$ is a 2-dimensional Gaussian random vector, with the prior mean and covariance given as

$$\mathbf{x}^b = [2.3, 2.5]^T \qquad \mathbf{B} = \begin{bmatrix} 0.225 & 0.05 \\ 0.05 & 0.15 \end{bmatrix}.$$

Only x_1 is observed, and the observed value is 3. Its error is normally distributed with a variance of 0.15.

$$[x|y] \propto [y|x][x]$$

w/ $[x] \sim N(x^B, B)$ and $[y|x] \sim [3, 0.15]$

$[x]$: prior; $[y|x]$: likelihood

Can use KF update (bayesian method)

with $H = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, only observe x_1, as $y = x_1$ observation

Then, can code:

$$K = P^- H (R + H P^- H^T)^{-1}$$

$$x^+ = x^- + K(y - H x^-)$$

$$P^+ = (I - KH) P^-$$

In Python, we get

```python
# Prior mean and covariance
x_prior = np.array([2.3, 2.5])
P_prior = np.array([[0.225, 0.05],[0.05, 0.15]])

# Measurement
y = 3.0
R = np.array([[0.15]])

# Observation mat - only observe x1
H = np.array([[1.0, 0.0]])

# Innovation z
z = y - H @ x_prior

# Kalman gain
K = P_prior @ H.T @ np.linalg.inv(H @ P_prior @ H.T + R)

# Posterior mean
x_posterior = x_prior + (K @ z).flatten()

# Posterior covariance
P_posterior = (np.eye(2) - K @ H) @ P_prior
```

Given $x^b = [2.3; 2.5]$ $B = \begin{bmatrix} 0.225, & 0.05 \\ 0.05 & 0.15 \end{bmatrix}$

and $y = 3$ w/ $R = 0.15$ and $H = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

We get an updated posterior

Correlated: $x^+ = \begin{bmatrix} 2.72 \\ 2.59\bar{3} \end{bmatrix}$ $P^+ = \begin{bmatrix} 0.09 & 0.02 \\ 0.02 & 0.14\bar{3} \end{bmatrix}$

2.) In comparison, if we use $P = \begin{bmatrix} 0.225 & 0 \\ 0 & 0.15 \end{bmatrix}$

We get:

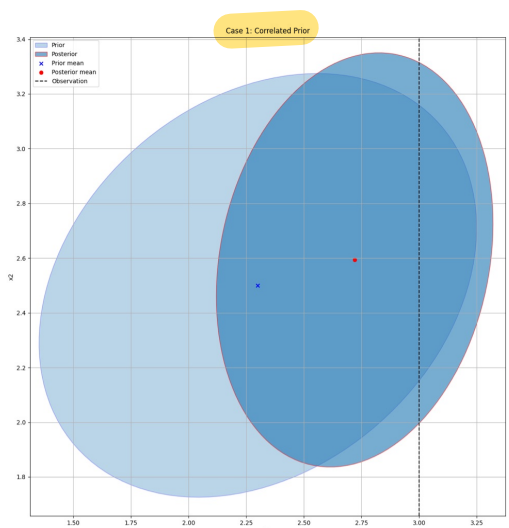
Uncorrelated:

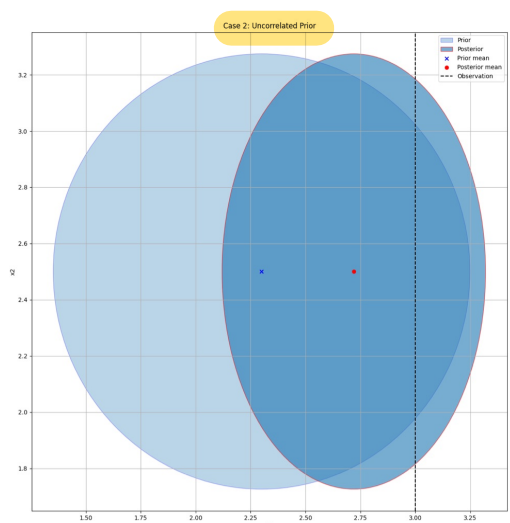
$$x^+ = \begin{bmatrix} 2.72 \\ 2.5 \end{bmatrix} \quad P^+ = \begin{bmatrix} 0.09 & 0 \\ 0 & 0.15 \end{bmatrix}$$

The core difference is that w/o cross correlation, the observation in x_1 does not update the x_2 term at all.

As before, we had cross correlation which causes impact to x_2 w/ the x_1 update.

Graphically we can compare the KF updates b/w correlated and uncorrelated





Very visible to see the linear shift when uncorrelated vs the interesting correlated shift that adjusts shape and orientation due to the update.